# Deep Belief Network を用いた検索用語の予測

馬　　青[†]・谷河　息吹[†]・村田　真樹[††]

本稿は機械学習を用いて関連語・周辺語または説明文書から適切な検索用語を予測する手法を提案する．機械学習には深層学習の一種である Deep Belief Network (DBN) を用いる．DBN の有効性を確認するために，用例に基づくベースライン手法，多層パーセプトロン (MLP)，サポートベクトルマシン (SVM) との比較を行った．学習と評価に用いるデータは手動と自動の 2 通りの方法でインターネットから収集した．加えて，自動生成した疑似データも用いた．各種機械学習の最適なパラメータはグリッドサーチと交差検証を行うことにより決定した．実験の結果，DBN の予測精度はベースライン手法よりはるかに高く MLP と SVM のいずれよりも高かった．また，手動収集データに自動収集のデータと疑似データを加えて学習することにより予測精度は向上した．さらに，よりノイズの多い学習データを加えても DBN の予測精度はさらに向上したのに対し，MLP の精度向上は見られなかった．このことから，DBN のほうが MLP よりもノイズの多い学習データを有効利用できることが分かった．

キーワード：深層学習，Deep Belief Network，検索用語予測，関連語，周辺語，情報検索支援

# Retrieval Term Prediction Using Deep Belief Networks

Qing Ma[†], Ibuki Tanigawa[†] and Masaki Murata[††]

This paper presents a method to predict retrieval terms from relevant/surrounding words or descriptive texts in Japanese by using deep belief networks (DBN), one of two typical types of deep learning. To determine the effectiveness of using DBN for this task, we tested it along with baseline methods using example-based approaches and conventional machine learning methods, i.e., multi-layer perceptron (MLP) and support vector machines (SVM), for comparison. The data for training and testing were obtained from the Web in manual and automatic manners. Automatically created pseudo data was also used. A grid search was adopted for obtaining the optimal hyperparameters of these machine learning methods by performing cross-validation on training data. Experimental results showed that (1) using DBN has far higher prediction precisions than using baseline methods and higher prediction precisions than using either MLP or SVM; (2) adding automatically gathered data and pseudo data to the manually gathered data as training data is an effective measure for further improving the prediction precisions; and (3) DBN is able to deal with noisier training data than MLP, i.e., the prediction precision of DBN can be improved by adding noisy training data, but that of MLP cannot be.

[†] 龍谷大学理工学部, Faculty of Science and Technology, Ryukoku University
[††] 鳥取大学工学研究科, Graduate School of Engineering, Tottori University

**Key Words**: *Deep Learning, Deep Belief Network, Retrieval Term Prediction, Relevant Word, Surrounding Word, Information Retrieval Support*

## 1　はじめに

Google に代表される現在の検索エンジンはその性能が非常によくなってきており，適切な検索用語（キーワード）さえ与えてやればおおむね期待通りの検索結果が得られる．しかし一方，多くのユーザ，特に子どもや高齢者，外国人などにとって検索対象を表す適切な検索用語（特に専門用語など）を見つけることは往々にしてそう簡単ではない．マイクロソフトの「現在の検索で不満に思う点」に関する調査[1]によれば，57.6%の人が適切なキーワード探しの難しさに不満を感じている．また，「何か欲しい情報を求めて検索エンジンを利用しているのに，それを利用するための適切なキーワードをまた別のところで探さねばならないという，堂々巡りをした経験を持つ人も多いはず」とも指摘されている．これは 2010 年の調査ではあるが，現在においてもこれらの不満点が大方解消されたとは言い難い．そこで，関連語・周辺語（たとえば「コンピュータ」，「前の状態」，「戻す」）またはそれらの語から構成される文を手掛かりに適切な検索用語（この場合「システム復元」）を予測・提示する検索支援システムがあればより快適な検索ができるのではないかと考えられる．

本研究では，IT や医療など様々な分野において，これらの分野の関連語・周辺語またはそれらの語から構成される文を入力とし，機械学習を用いて適切な検索用語を予測・提示する検索支援システムの開発を目標としている．このような研究は，すくなくとも日本語においては我々が調べた限りではこれまでなされていなかった[2]．本稿ではその第一歩として，分野をコンピュータ関連に限定し，深層学習 (Deep Learning) の一種である Deep Belief Network (DBN) を用いた予測手法を提案する．

近年，深層学習は様々な分野で注目され，音声認識 (Li, Zhao, Jiang, Zhang, Wang, Gonzalez, Valentin, and Sahli 2013) や画像認識 (Krizhevsky, Sutskever, and Hinton 2012) のみならず，自然言語処理の諸課題への応用にも優れた性能を出している．それらの諸課題は，形態素・構文解析 (Billingsley and Curran 2012; Hermann and Blunsom 2013; Luong, Socher, and Manning 2013; Socher, Bauer, Manning, and Ng 2013)，意味処理 (Hashimoto, Miwa, Tsuruoka, and Chikayama 2013; Srivastava, Hovy, and Hovy 2013; Tsubaki, Duh, Shimbo, and Matsumoto

[1] http://www.garbagenews.net/archives/1466626.html または http://news.mynavi.jp/news/2010/07/05/028/

[2] 類似研究として，「意味的逆引き辞書」に関する研究 (栗飯原，長尾，田中 2013) や「クロスワードを解く」に関する研究 (内木，佐藤，駒谷 2013) がある．しかしこれらは分野ごとの検索用語の予測・提示に基づく検索支援を第一の目的としておらず，それゆえに，精度（正解率）は本研究で得られたものよりはるかに低かった．また，手法も LSI を利用した情報検索技術やエキスパートなどに基づくアプローチを取っており，本研究が取っている機械学習のアプローチとは異なる．

2013)，言い換え (Socher, Huang, Pennington, Ng, and Manning 2011)，機械翻訳 (Auli, Galley, Quirk, and Zweig 2013; Liu, Watanabe, Sumita, and Zhao 2013; Kalchbrenner and Blunsom 2013; Zou, Socher, Cer, and Manning 2013)，文書分類 (Glorot, Bordes, and Bengio 2011)，情報検索 (Salakhutdinov and Hinton 2009)，その他 (Seide, Li, and Yu 2011; Socher, Perelygin, Wu, and Chuang 2013) を含む．さらに，統一した枠組みで品詞タグ付け・チャンキング・固有表現認識・意味役割のラベル付けを含む各種の言語処理課題を取り扱えるニューラルネットおよび学習アルゴリズムも提案されている (Collobert, Weston, Bottou, Karlen, Kavukcuoglu, and Kuksa 2011).

しかしながら，われわれの知っている限りでは，前に述べたような情報検索支援に関する課題に深層学習を用いた研究はこれまでなされていない．したがって，本稿で述べる研究は主に二つの目的を持っている．一つは，関連語・周辺語などから適切な検索用語を正確に予測する手法を提案することである．もう一つは，深層学習がこのような言語処理課題において，従来の機械学習手法である多層パーセプトロン (MLP) やサポートベクトルマシン (SVM) より優れているか否かを確かめることである．

本研究に用いたデータはインターネットから精度保証がある程度できる手動収集と，ノイズ[3]は含まれるが規模の大きいデータの収集が可能な自動収集との 2 通りの方法で収集した．加えて，ある程度規模が大きく精度もよい疑似データも自動生成して用いた．機械学習のパラメータチューニングはグリッドサーチと交差検証を用いて行った．実験の結果，まず，学習データとして手動収集データのみを用いても自動収集データと疑似データを加えても DBN の予測精度は用例に基づくベースライン手法よりははるかに高く MLP と SVM のいずれよりも高いことが確認できた．また，いずれの機械学習手法も，手動収集データにノイズの多い自動収集データとノイズの少ない疑似データを加えて学習することにより予測精度が向上した．さらに，手動収集データにノイズの多い自動収集データのみを加えて学習した場合，DBN と SVM には予測精度の向上が見られたが MLP にはみられなかった．この結果から，MLP よりも DBN と SVM のほうがノイズに強くノイズの多い学習データも有効利用できる可能性が高いと言えよう．

## 2 関連語・周辺語コーパス

機械学習を用いて関連語・周辺語から検索用語を予測・提示する場合，その学習データとして，入力（関連語・周辺語）と正解となるレスポンス（検索用語）のペアからなるコーパスが必要となる．本稿ではこのようなコーパスを「関連語・周辺語コーパス」と呼ぶ．また，教師あり機械学習では，レスポンスをラベルと呼ぶ場合が多いので，本稿では検索用語をラベルと

[3] ここのノイズとは，関係のない単語が含まれている，または必要な単語が欠落していることを指す．

呼ぶ．表 1 はコーパスの入力（関連語・周辺語とその元となる説明文書）とラベルのペアの例を示す．本章では，コーパスデータの収集・作成方法について述べる．また，収集・作成したデータからの関連語・周辺語の抽出方法と特徴ベクトルの構成方法について述べる．

**表 1**　コーパスの入力とラベルのペアの例

| ラベル | | 入力 |
|---|---|---|
| グラフィックボード | 説明文書 | 別名：グラフィックカード，グラフィックアクセラレーター，グラボ，VGA．実際に目で見る画面を映し出しているのはディスプレイですが，ディスプレイは命令通りに表示しているだけに過ぎず，命令が無ければ何も映りません．その命令を出す機械が，グラフィックボードです．グラフィックボードがパソコン上で処理する必要があるデータは 2 種類存在し，「2D（平面）と 3D（立体）」です．2D におけるグラフィック性能は，複雑な処理を要求される事がないために極端に古くなければ気にかける必要はありません．つまり，グラフィックボードで性能に差が現れる部分は「3D 処理能力」と言えます．これは，グラフィックボードにより処理速度に大きな差があります． |
| | 関連語・周辺語 | 別名，グラフィックカード，グラフィックアクセラレーター，グラボ，VGA, 目，画面，ディスプレイ，命令通り，表示，命令，機械，パソコン上，処理，データ，2 種類存在，2D, 平面，3D, 立体，グラフィック性能，要求，気，性能，差，部分，3D 処理能力，処理速度 |
| | 説明文書 | パーソナルコンピュータなどの各種のコンピュータで，映像を信号として出力または入力する機能を，拡張カード（拡張ボード）として独立させたもの．カードに搭載されているチップやメモリによって描画速度，解像度，3D 性能などが異なる． |
| | 関連語・周辺語 | パーソナルコンピュータ，各種，コンピュータ，映像，信号，出力，入力，機能，拡張カード，拡張ボード，独立，カード，搭載，チップ，メモリ，描画速度，解像度，3D 性能 |
| メインメモリ | 説明文書 | コンピュータ内でデータやプログラムを記憶する装置．「一次記憶装置」とも呼ばれる．半導体素子を利用して電気的に記録を行うため，動作が高速で，CPU（中央処理装置）から直接読み書きすることができるが，単位容量あたりの価格が高いため大量には使用できない．また，電源を切ると内容が失われてしまうという欠点がある．このため，コンピュータにはメインメモリのほかに，ハードディスクやフロッピーディスクなどの外部記憶装置（補助記憶装置）が装備されており，利用者がプログラムを起動してデータの加工を行う際には必要なものだけメインメモリに呼び出して使い，長期的な保存には外部記憶装置が利用される． |
| | 関連語・周辺語 | コンピュータ内，データ，プログラム，記憶，装置，一次記憶装置，半導体素子，利用，電気的，記録，動作，高速，CPU, 中央処理装置，直接読み書き，単位容量あたり，価格，ため大量，使用，電源，内容，欠点，コンピュータ，メインメモリ，ハードディスク，フロッピーディスク，外部記憶装置，補助記憶装置，装備，利用者，起動，加工，長期的，保存 |
| | 説明文書 | メインメモリとは，パソコンのデータを一時的に記憶しておく装置のことです．パソコンの性能を上げるためにはこのメインメモリの容量を多くすることが重要です． |
| | 関連語・周辺語 | メインメモリ，パソコン，データ，一時的，記憶，装置，性能，容量 |

## 2.1 手動収集と自動収集

本研究では，ラベルを説明している文書には関連語・周辺語が多く含まれると考え，インターネットからこのような Web ページを手動と自動の 2 通りの方法で収集した．手動収集では人手でラベルを説明する Web ページを選別し収集する[4]．一方，自動収集では，ラベルの後に「とは」「は」「というものは」「については」「の意味は」の 5 語を付けて（たとえば，ラベルが「グラフィックボード」であれば「グラフィックボードとは」「グラフィックボードというものは」などで）Google で検索したものを説明文書として収集する[5]．手動収集データは規模が小さい代わりに精度が高く，自動収集データは精度が低い代わりに規模が大きい．

## 2.2 疑似データ

機械学習の汎化能力を向上させるために，学習データとして，精度は高いが規模が小さい手動収集データに加え，精度はそれほど高くない（つまり，ノイズはある）が相対的に規模の大きい自動収集データを用いることにした．しかし，自動収集したデータには説明文書とラベルがそもそも一致しない，つまり説明文書へのラベルが履き違えられている可能性も考えられる．そのために，手動で収集した説明文書をオリジナルのデータとしてとらえ，それらに適度なノイズを加えて作成した疑似データも用いることにした．このようなデータは自動収集したデータに比べノイズが少なくラベルの履き違いもないと考えることができる．疑似データの具体的な生成手順は以下の通りである．

(1) オリジナルの説明文書からすべての異なり単語を抽出する．
(2) 個々のオリジナルの説明文書に対し，追加，削除，または追加&削除の処理を加える．具体的には，手順 (1) で抽出した単語のうち，説明文書にない単語を説明文書の単語数の 10%個ランダムに選んで加える，説明文書から単語を説明文書の単語数の 10%個ランダムに選んで削除する，または上記の（10%ずつの）追加と削除を同時に施す，という処理を等確率（つまり，それぞれを 1/3 の確率）で行う[6]．
(3) 手順 (2) で得られたデータを疑似データとする．

なお，この生成方法においては，1 つのオリジナルの説明文書に対し，疑似データを複数生成することが可能である．

[4] 人手で Web ページを選別した後，その Web ページから説明文書として該当する箇所を人手で選別する処理を行っている．
[5] 収集した Web ページ全体をそのラベルの説明文書として扱っている．
[6] 10%という値は，予備実験などで精査して決めたものではなく，著者らが適度なノイズとして主観で設定したものである．

## 2.3 評価データ

評価データは学習データとは別に自動収集したものを用いる．ただし，自動収集データは，ラベルが正確とは限らないため，評価データとして用いても適切な評価とならない可能性がある．そのため，評価データとして自動収集データの中からラベルの正しいものを人手で選別して用いることにした．

## 2.4 関連語・周辺語抽出とベクトル変換

以下の手順 (1)〜(4) で説明文書から関連語・周辺語を抽出する．それに手順 (5)(6) を加えることにより，機械学習に必要な特徴ベクトルへの変換を行う．

(1) 手動収集のデータを形態素解析し，名詞（固有名詞，サ変接続，一般）を抽出する[7]．

(2) 名詞が連続しているならば，日本語同士なら結合し，英語同士なら空白を間に入れて結合し，1 つの単語と見なす．

(3) 各ラベルから出現頻度がトップ 50 以内の単語を抽出する[8]．

(4) ラベル間で重複している単語を除外する．本研究では，以下に述べる考えに基づき 2 ラベル間で重複する単語を除外する，または，3 ラベル以上で共通する単語を除外するという 2 通りの方法を採用した．まず，各ラベルにできるだけ特徴的な単語のみを素性にするためには重複単語をできるだけ除外するのが効果的と考える．また，今回は実験規模が小さくあまり問題にならないが，予測用語の数の増加に伴う特徴ベクトル次元の大幅な増加を抑える 1 つの方法として重複単語を除外することが考えられる．特徴ベクトル次元の抑制はまた一般的に，学習におけるデータスパースネス問題の緩和にもつながる．しかし一方，ラベル間の単語重複をまったく認めないと，たとえば「USB メモリ」のような，「USB」や「メモリ」に共通する重要な単語を除外してしまう問題も考えられる．そのため，本研究では 2 ラベル間の重複を許容し 3 ラベル以上で共通する単語を除外する方法も用いる．

(5) 上記手順で得られた単語をベクトルの要素とし，個々の要素はその単語が出現していれ

[7] 形態素解析に MeCab 0.98 を使用した．未知語と表記ゆれについては特別な処理を施しておらず今後の課題となる．ただし，未知語としての複合語については，たとえば「木村製作所」や「株式会社ウエーブ」など大半の日本企業名の場合は（中小企業で社名としては未知語であっても）「木村」と「製作所」などがそれぞれ名詞として解析されているので，手順 (2) にしたがって問題なく 1 つの既知単語として扱われる．一方，たとえば「騰迅公司」のような表現において，最初の漢字が名詞以外の品詞と判断された場合は 1 つの既知単語として正しく扱うことができない．また，表記ゆれについては，「サーバ」と「サーバー」，「神経回路」と「ニューラルネット」のような形態素解析ツールの辞書に登録されているものはそれぞれ異なる単語として扱われてしまい，予測性能を落とす可能性がある．

[8] 50 という値は，手動で収集したデータにおいて各ラベルの関連語・周辺語の数は確実にそれ以下であることを確認した上で，提案手法の拡張性（つまり多少大きい目に）と機械学習の素性選択能力（つまり多少大きい目にしても問題がないこと）も考慮にいれて設定した．なお，この値は多少大きく設定されても手順 (4) で絞られるので値をある程度大きい目に設定しておけば 40 がよいか 60 がよいかといった細かい選択はほとんど意味をなさないと思われる．

ば 1，出現していなければ 0 の 2 値を取る．

(6) 2.1, 2.2, 2.3 節で述べたすべてのデータに対し形態素解析を行い，手順 (5) にしたがって特徴ベクトルに変換する．

## 3 深層学習

深層学習とは従来の機械学習より深い層構造をしている機械学習手法全般のことを指す．その代表的な手法として Deep Belief Network (DBN) (Hiton, Osindero, and Teh 2006; Lee, Grosse, Ranganath, and Ng 2009; Bengio 2009; Bengio, Courville, and Vincent 2013) と Stacked Denoising Autoencoder (SdA) (Bengio, Lamblin, Popovici, and Larochelle 2007; Bengio 2009; Bengio et al. 2013; Vincent, Larochelle, Bengio, and Manzagol 2008; Vincent, Larochelle, Lajoie, Bengio, and Manzagol 2010) が提案されている．数多くの課題において，その両者の性能がほぼ同じと言われているが，本研究ではよりスマートなアーキテクチャを有する DBN を用いることにした．

深層学習は，本来経験則で行っていた特徴抽出を機械学習に組み込もうとしてできたものである．そのため，DBN は，Restricted Boltzmann Machine (RBM) を複数並べ教師なし学習の特徴抽出器として利用する多層のニューラルネットと，ラベルを出力する教師あり学習の最終層から構成される．特徴抽出器の教師なし学習は Pre-training，最終層の教師あり学習は Fine-tuning と呼ばれる．

### 3.1 Restricted Boltzmann Machine (RBM)

RBM は制限付きボルツマンマシンとも呼ばれ，学習データの確率分布を教師なし学習で表現する（言い換えれば，学習データの生成モデルを統計的な機械学習の方法で構築する），一種の確率的なグラフィカルモデルである．本来のボルツマンマシンの可視層と隠れ層のユニット間の結合を制限することにより，効率的な教師なし学習を実現している．

RBM の構造は図 1 に示しているように可視層と隠れ層の 2 層から構成され，層内ユニット間に結合がなく，層間のユニット，すなわち可視ユニット $(v_1, v_2, \cdots, v_m)$ と隠れユニット $(h_1, h_2, \cdots, h_n)$，は結合されている．

以下，その学習アルゴリズム (Bengio 2009) を簡潔に述べておく．

学習データ $\boldsymbol{v}$ が可視層に与えられたとき，まず，式 (1)，(2)，そして再度 (1) の順で条件付確率に基づくサンプリングを行う．

$$P(h_i^{(k)} = 1|\boldsymbol{v}^{(k)}) = \text{sigmoid}\left(\sum_{j=1}^{m} w_{ij} v_j^{(k)} + c_i\right) \tag{1}$$

$$P(v_j^{(k+1)} = 1|\boldsymbol{h}^{(k)}) = \text{sigmoid}\left(\sum_{i=1}^{n} w_{ij} h_i^{(k)} + b_j\right) \tag{2}$$

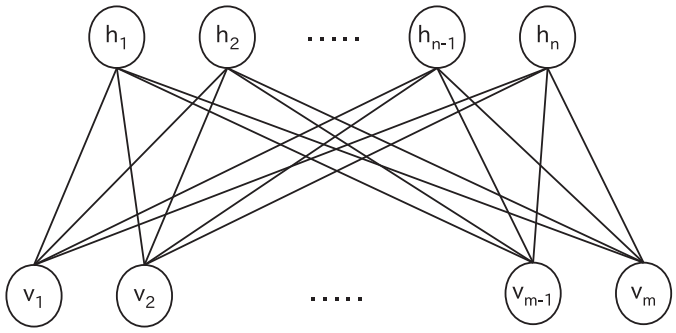



**図 1**　Restricted Boltzmann Machine の構造

ただし，$k$ ($\geq 1$) はサンプリングの繰り返し回数，$\boldsymbol{v}^{(1)} = \boldsymbol{v}$，$w_{ij}$ はユニット $v_j$ と $h_i$ 間の結合の重み，そして，$b_j$ と $c_i$ は可視層と隠れ層のユニット $v_j$ と $h_i$ のオフセット（バイアス）である．サンプリングを $k$ 回行った後，重みとオフセットは以下のように更新される．

$$\boldsymbol{W} \leftarrow \boldsymbol{W} + \epsilon(\boldsymbol{h}^{(1)}\boldsymbol{v}^T - P(\boldsymbol{h}^{(k+1)} = 1|\boldsymbol{v}^{(k+1)})\boldsymbol{v}^{(k+1)T}) \tag{3}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} + \epsilon(\boldsymbol{v} - \boldsymbol{v}^{(k+1)}) \tag{4}$$

$$\boldsymbol{c} \leftarrow \boldsymbol{c} + \epsilon(\boldsymbol{h}^{(1)} - P(\boldsymbol{h}^{(k+1)} = 1|\boldsymbol{v}^{(k+1)})) \tag{5}$$

ただし，$\epsilon$ は学習率である．$\boldsymbol{W}$ は微小な乱数[9]，$\boldsymbol{b}, \boldsymbol{c}$ は $\mathbf{0}$ で初期化する．サンプリングの繰り返し回数が十分多いときは Gibbs sampling と呼ばれており計算コストが非常に高い．そのため，通常，サンプリングを $k$ 回のみ行う $k$-Contrastive Divergence （略して CD-$k$）と呼ばれる方法が採用される．実際，$k = 1$ (CD-1) でも結果が十分よいことが経験的に知られており (Bengio 2009)，本研究も $k = 1$ に設定して学習を行う．

ここで $N$ 個の学習データに対し CD-$k$ と呼ばれるサンプリング方法で $e$ 回繰り返し学習を行う手順を図 2 にまとめる．学習が進むにつれ，可視層のサンプル[10] $\boldsymbol{v}^{(k+1)}$ が学習データ $\boldsymbol{v}$ に近づいていく．

## 3.2　Deep Belief Network (DBN)

図 3 は一例として，三つの RBM と教師あり学習器から構成される DBN を示す．ただし実際，DBN を構成する RBM の数は可変である．それら RBM は Pre-training とも呼ばれ，教師なしの特徴抽出器として機能する．一方，教師あり学習器は Fine-tuning とも呼ばれ，入力（図 3 の場合はその入力から得られた RBM 3 の出力）とラベルのペア（つまり正解付学習データ）を

[9] 本研究では，http://deeplearning.net/tutorial/mlp.html のチュートリアルに従って，区間 $[-4\frac{\sqrt{6}}{\sqrt{m+n}}, 4\frac{\sqrt{6}}{\sqrt{m+n}}]$ 内の一様乱数を用いる（ただし，$m$ と $n$ はそれぞれ可視層と隠れ層のユニット数である）．その数学的な考えについては (Glorot and Bengio 2010) を参照されたい．

[10] ここでは条件付確率の式 (1)(2) に基づき生成されたデータをサンプルと呼んでいる．

学習することにより未知の入力に対しても適切なラベルを出力できるようになる．図に示しているように前方の RBM の隠れ層は後方の RBM の可視層となっている．ここでは簡便化のために，RBM の層（ただし入力層を除く）を DBN の隠れ層と見なす．つまり，図の例は三層の隠れ層の DBN である（隠れ層の数と RBM の数は同じであることに注意されたい）．なお，教師あり学習はいろいろな方法で実現できるが，本稿ではロジスティク回帰を用いることにした．

三つの RBM を持つ DBN の学習手順を図 4 にまとめる．

(1) 学習の回数が $e$ に達するまで以下を繰り返す
  (1.1) $N$ 個の学習データの個々について以下を繰り返す
    (1.1.1) 1 個の学習データ $\boldsymbol{v}$ に対し以下の処理を $k$ 回繰り返す（つまり CD-$k$ サンプリング）
      (1.1.1.1) 式 (1), (2), (1) に基づきサンプリングする
    (1.1.2) 式 (3), (4), (5) に基づき重みとオフセットを更新する

**図 2**　RBM の学習手順

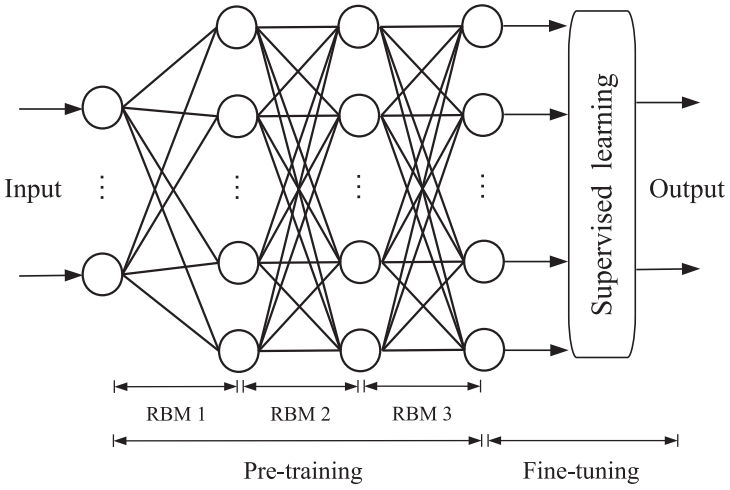



**図 3**　Deep Belief Network の例

(1)　学習データを入力として **RBM の学習手順（図 2）** にしたがって RBM 1 を訓練し，訓練により更新された重みとオフセットをフィックスする
(2)　RBM 1 の隠れ層のサンプルを入力として **RBM の学習手順（図 2）** にしたがって RBM 2 を訓練し，訓練により更新された重みとオフセットをフィックスする
(3)　RBM 2 の隠れ層のサンプルを入力として **RBM の学習手順（図 2）** にしたがって RBM 3 を訓練し，訓練により更新された重みとオフセットをフィックスする
(4)　教師あり学習器に対し，RBM 3 の隠れ層のサンプルと正解とのペアを用いて教師あり学習を行う

**図 4**　三つの RBM を持つ DBN の学習手順

# 4 実験

## 4.1 実験設定

### 4.1.1 データ

学習と評価には 10 個のラベルとそれらの入力（説明文書）のペアから構成されるデータを用いた．表 2 はラベル名と各ラベルの入力（説明文書）の数とそれらの全ラベルに占める割合を示す[11]．ただし，学習データは，手動収集データをベースとし，そのベースとなるデータに異なる数の自動収集データと疑似データを加えることにより 13 個のデータセット（表 3）を作成して用いた．表中の m300 はベースとなるデータセットで手動で収集した 300 個のデータである．また，たとえば a2400 は 2,400 個の自動収集データと m300 で構成されたデータセット，p2400 は 2,400 個の疑似データと m300 から構成されたデータセット，そして a2400p2400 は 2,400 個の自動収集データ，2,400 個の疑似データ，そして m300 から構成されたデータセットである．また，評価には学習データと異なる 100 個のデータを用いた． 個々の説明文書は 2.4 節で述べた方法で，2 ラベル間で重複する単語を除外する場合と 3 ラベル以上で共通する単語を除外する場合においてそれぞれ 182 と 223 次元の特徴ベクトルに変換される．

**表 2** 10 個のラベルと各ラベルの入力（説明文書）の数とそれらの全ラベルに占める割合

| | 学習データ | | | | 評価データ |
|---|---|---|---|---|---|
| | 手動 | 疑似 | 自動 | 合計 | 自動 |
| CPU | 30 (0.1) | 240 (0.1) | 302 (0.125) | 572(0.112) | 10 (0.1) |
| グラフィックボード | 30 (0.1) | 240 (0.1) | 203 (0.084) | 473(0.093) | 10 (0.1) |
| ハードディスク | 30 (0.1) | 240 (0.1) | 367 (0.152) | 637(0.125) | 10 (0.1) |
| メインメモリ | 30 (0.1) | 240 (0.1) | 214 (0.089) | 484(0.095) | 10 (0.1) |
| マザーボード | 30 (0.1) | 240 (0.1) | 268 (0.111) | 538(0.105) | 10 (0.1) |
| OS | 30 (0.1) | 240 (0.1) | 288 (0.120) | 558(0.109) | 10 (0.1) |
| 光学ドライブ | 30 (0.1) | 240 (0.1) | 222 (0.092) | 492(0.096) | 10 (0.1) |
| PC ケース | 30 (0.1) | 240 (0.1) | 188 (0.078) | 458(0.090) | 10 (0.1) |
| 電源ユニット | 30 (0.1) | 240 (0.1) | 153 (0.063) | 423(0.083) | 10 (0.1) |
| SSD | 30 (0.1) | 240 (0.1) | 195 (0.081) | 465(0.091) | 10 (0.1) |

**表 3** 学習用データセット

| m300 | | | |
|---|---|---|---|
| a300 | a600 | a1200 | a2400 |
| p300 | p600 | p1200 | p2400 |
| a300p300 | a600p600 | a1200p1200 | a2400p2400 |

[11] 自動収集データにおいて各ラベルのデータ数に多少のバラつきがあるが，ある程度のバランスが取れている．

#### 4.1.2 パラメータのチューニング

各種の機械学習の各学習データセットにおける最適なパラメータは，それぞれの学習データセットに対しグリッドサーチと 5-fold 交差検証を行って決定した．グリッドサーチに用いるパラメータの詳細は表 4 にまとめている．たとえば，DBN の入力が 182 次元の場合の構造（隠れ層）の欄に 152-121-91 がある．これは，その DBN は 182-**152-121-91**-10 という構造を持つ，ということを表している．ただし，数字 182 と 10 は入力層と出力層のユニット数であり，それぞれ特徴ベクトルの次元数とラベルの数に対応している．また，これら隠れ層のユニット数は恣意的にではなく，前半の 3 つについては線形等間隔に設定している．すなわち，入力層のユニット数 (182) から，ピラミッド的に，最初の隠れ層のユニット数を $182 \times 5/6$ (152)，次の隠れ層のユニット数を $182 \times 4/6$ (121)，そして，最後の隠れ層のユニット数を $182 \times 3/6$ (91) のように設定している．一方，後半の 3 つについては，Bengio の (Bengio 2012) の薦め，すなわち，過学習への対処が適切であれば隠れ層のユニット数は基本的に多いほどよい，ネットワー

**表 4** グリッドサーチに用いるパラメータ

| 機械学習 | パラメータ | 値 |
|---|---|---|
| DBN | 入力が 182 次元の場合の構造（隠れ層） | 91, 137-91, 152-121-91, 273, 273-273, 273-273-273 |
| | 入力が 223 次元の場合の構造（隠れ層） | 112, 167-112, 186-149-112, 335, 335-335, 335-335-335 |
| | Pre-training の学習率 | 0.001, 0.01, 0.1 |
| | Fine-tuning の学習率 | 0.001, 0.01, 0.1 |
| | Pre-training の学習回数 | 500, 1,000, 2,000, 3,000 |
| | Fine-tuning の学習回数 | 500, 1,000, 2,000, 3,000 |
| MLP 1 | 入力が 182 次元の場合の構造（隠れ層） | 91, 137-91, 152-121-91, 273, 273-273, 273-273-273 |
| | 入力が 223 次元の場合の構造（隠れ層） | 112, 167-112, 186-149-112, 335, 335-335, 335-335-335 |
| | 学習率 | 0.001, 0.01, 0.1 |
| | 学習回数 | 500, 1,000, 2,000, 3,000 |
| MLP 2 | 入力が 182 次元の場合の構造（隠れ層） | 91, 137-91, 152-121-91, 273, 273-273, 273-273-273 |
| | 入力が 223 次元の場合の構造（隠れ層） | 112, 167-112, 186-149-112, 335, 335-335, 335-335-335 |
| | 学習率 | 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025, 0.05, 0.075, 0.1 |
| | 学習回数 | 500–1,000 間に 6 等分割，1,200–3,000 間に 10 等分割 |
| SVM (Linear) | C | $10^{-4}$–$10^{4}$ 間に対数（基底 10）スケールで 900 分割 |
| SVM (RBF) | $\gamma$ | $10^{-4}$–$10^{4}$ 間に対数（基底 10）スケールで 30 分割 |
| | C | $10^{-4}$–$10^{4}$ 対数（基底 10）スケールで 30 分割 |

ク構造は各層が同じサイズでよい場合が多い（ピラミッドまたは逆ピラミッドである必要はない）に基づき，すべての隠れ層のユニット数を入力層のユニット数の 3/2 倍であるように設定した．入力層のユニット数が 223 の場合も同様な考え方に基づいて設定した．

DBN が MLP と SVM よりパラメータが多いため，同じ細かさのグリッドサーチで最適なパラメータを決めてしまうと，パラメータの多い DBN のほうが細かなチューニングができるため有利になる可能性がある．このようなバイアスをなくすために，MLP と SVM についてそのパラメータグリッドをより細かくし，MLP と SVM の探索すべきパラメータセットの数（つまり，パラメータの組み合わせの数）を DBN のそれと等しいかそれ以上にした．一方，MLP については，構造，学習率，学習回数が DBN とまったく同じものも比較に用いた．本稿では後者を MLP 1，前者を MLP 2 と呼ぶ．その結果，DBN と MLP 2 は同じく 864 通りのパラメータセット，SVM (Linear) と SVM (RBF) は 900 通りのパラメータセット，また，MLP 1 は 72 通りのパラメータセットを持つことになる．

#### 4.1.3　ベースライン

MLP と SVM に加え，用例に基づく手法をベースラインとして比較実験に加えた．これは，評価データを学習データの一つひとつと比較し，共通する単語のもっとも多い，または共通する単語数をその評価データの単語数で正規化した値がもっとも大きい学習データのラベルを評価データのラベルとする方法である．ここで両者をそれぞれ Baseline 1 と Baseline 2 と呼ぶ．図 5 は本手法および本手法による予測結果の正解率算出のアルゴリズムを示す．ただし，カウントに用いる単語は 2.4 節で述べた (1)〜(4) の手順に従って説明文書から抽出されたものである．

(1) i 番目の評価データの入力について
　(1.1) j 番目の学習データの入力について
　　(1.1.1) i との共通する単語をカウントする
　(1.2) j = N-learn　まで (1.1) を繰り返し，
　　Baseline 1 の場合，共通する単語のもっとも多い j を見つけ，m = j とする
　　Baseline 2 の場合，共通する単語数をその評価データの単語数で
　　割った値がもっとも大きい j を見つけ，m = j とする
　　ただし N-learn：学習データの総数
　(1.3) m 番目の学習データのラベルを i 番目の評価データの予測結果 r とする
　(1.4) r と i 番目の評価データの正解ラベルとを比較し，正否判定をし，
　　正しい予測結果 N-correct をカウントする
　(1.5) i = N-test まで (1) を繰り返す．ただし N-test：評価データの総数
(2) 正しい予測結果のカウント N-correct を N-test で割り，正解率（精度）を算出する

**図 5**　ベースライン手法 (Baseline 1, Baseline 2) およびその正解率算出のアルゴリズム

## 4.2 実験結果

### 4.2.1 182 次元の特徴ベクトルを使用した場合

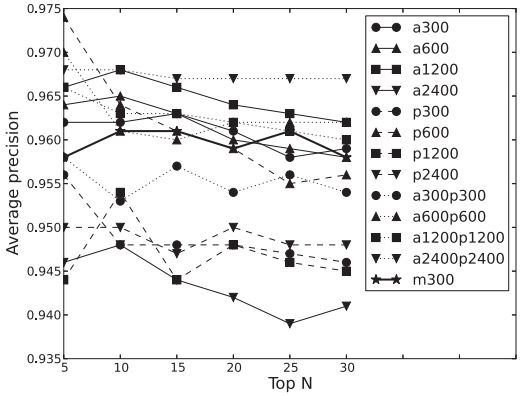
図 6 は各機械学習において，異なる学習データセットを用いた場合の評価データへの予測精度を示す．ここでの精度は，各パラメータセットの交差検証誤差を昇順（小さい順）に並べたときの上位 N 個（ただし N は 5 から 30 まで可変）のパラメータセットを用いた場合の平均精度である．なお，本論文に用いられている平均精度はすべてマクロ平均で算出したものである．

図に示しているように，全般的に見れば，学習データセット a2400p2400 を用いた場合（逆三角形マークの点線[12]），すなわち手動収集データに自動収集データと疑似データの両方を最も多く加えた場合，DBN と MLP は最高の精度，そして SVM もほぼ最高の精度を出している[13]．また，手動収集データに自動収集データと疑似データの両方を適度に加えた場合（点線）は，手動収集データのみの場合（星マークの太線）に比べ，DBN と MLP と SVM (RBF) の予測精度はおおむね向上している．しかし SVM (Linear) についてはそのような傾向は見られなかった[14]．さらに，手動収集データのみを用いた場合と，自動収集データと疑似データのどちらか一方のみを手動収集データに加えた場合について比べると，DBN と SVM (RBF) については自動収集データのみを加えた場合（実線），MLP については疑似データのみを加えた場合（破線）のほうがそれぞれに精度の向上が見られた．自動収集データのほうが疑似データよりもノイズが多いことから，上記結果は DBN と SVM (RBF) のほうが MLP よりもノイズの多い学習データを有効利用できる可能性が高いことを示している．

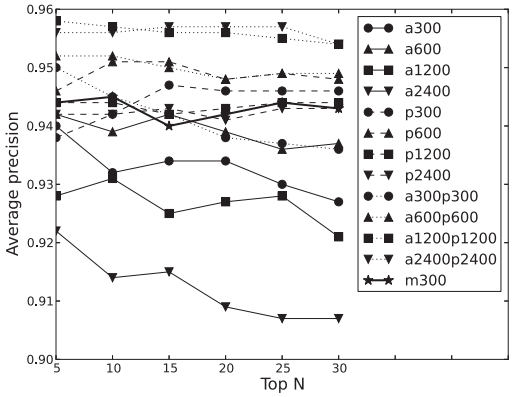
図 7 は各機械学習間の評価データへの予測精度の比較を示す．ここでの精度は図 6 と同様，各パラメータセットの交差検証誤差を昇順に並べたときの上位 N 個（ただし N は 5 から 30 まで可変）のパラメータセットを用いた場合の平均精度である．学習データセットも図 6 のとまったく同じであるがそれらの詳細の明示は省略されている．ただし，各グラフの縦軸の範囲が統一されているため，グラフ DBN vs. SVM (RBF) において，SVM (RBF) の精度が 0.9 未満なもの（計 4 本の線）が表示されていない（なお，すべての結果は図 6 には示されている）．この図からは DBN のほう（実線）が他の機械学習（破線）より性能がよいことが一目瞭然にわかる．

表 5，6，7，8 はそれぞれ，各学習データセットを用いた場合の，ベースラインの予測精度，交差検証誤差が最小のパラメータセットを用いた場合の予測精度，交差検証誤差を昇順に並べたときの上位 5 個，10 個のパラメータセットを用いた場合の平均予測精度を示す．まず，機械学習とは対照的に，ベースライン手法では，ノイズの多い学習データを加えても（つまり，手動収集データに自動収集データのみを加えた場合と，自動収集データと疑似データの両方を加えた場合），予測精度の向上に役立たないばかりか，逆に，これらのデータは予測精度を大きく

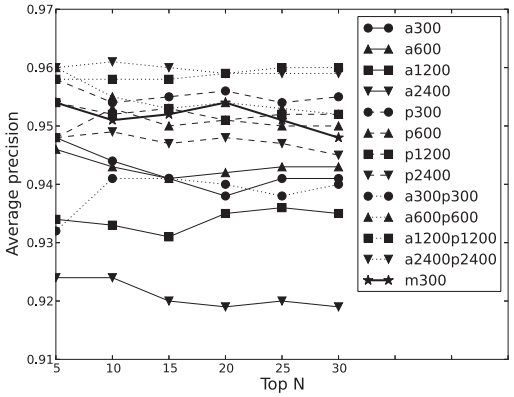
[12] 点線と破線の違いに注意されたい．
[13] SVM (RBF) の場合，逆三角形マークの点線は四角形マークの点線と重ねていることに注意されたい．
[14] これは SVM (Linear) が線形分離可能なデータしか取り扱えないことに起因するものと思われる．



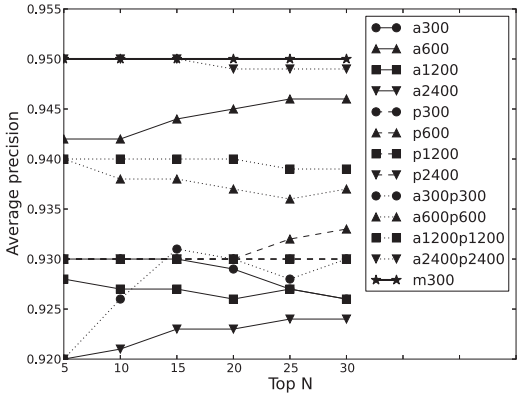

DBN



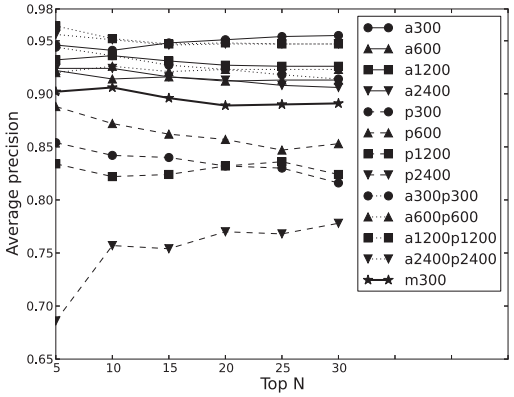

MLP 1




MLP 2

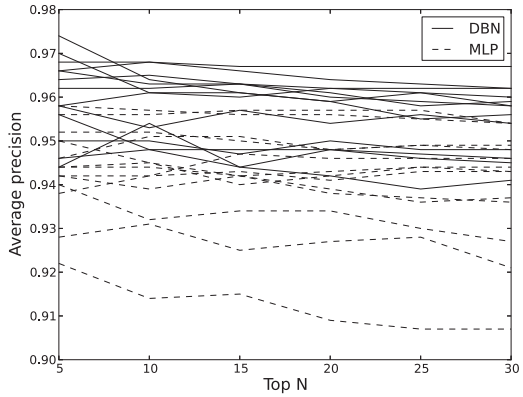



SVM (Linear)

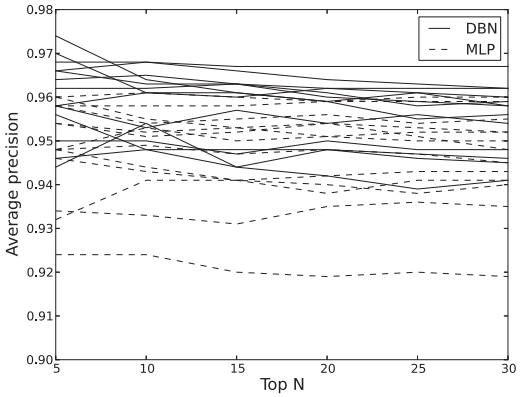



SVM (RBF)

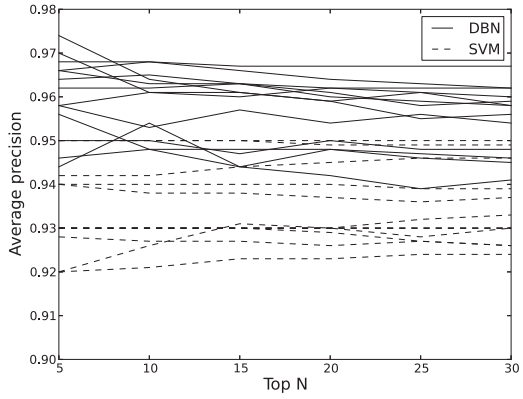
**図 6**　交差検証誤差の昇順で上位 N(5〜30) セットのパラメータを用いた場合の各機械学習の平均精度についての学習データセット間の比較（182 次元の特徴ベクトルを用いた場合）



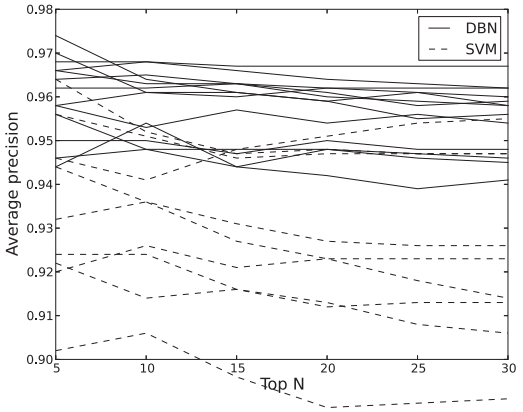

DBN vs. MLP 1




DBN vs. MLP 2




DBN vs. SVM (Linear)




DBN vs. SVM (RBF)

**図 7**　交差検証誤差の昇順で上位 N(5〜30) セットのパラメータを用いた場合の平均精度についての機械学習間の比較（182 次元の特徴ベクトルを用いた場合）

**表 5**　ベースラインの精度

| | m300 | a300 | a600 | a1200 | a2400 | p300 | p600 |
|---|---|---|---|---|---|---|---|
| Baseline 1 | 0.85 | 0.50 | 0.32 | 0.39 | 0.37 | 0.85 | 0.84 |
| Baseline 2 | 0.55 | 0.50 | 0.34 | 0.28 | 0.38 | 0.64 | 0.50 |

| | p1200 | p2400 | a300p300 | a600p600 | a1200p1200 | a2400p2400 |
|---|---|---|---|---|---|---|
| Baseline 1 | 0.84 | 0.84 | 0.51 | 0.32 | 0.39 | 0.37 |
| Baseline 2 | 0.62 | 0.53 | 0.41 | 0.36 | 0.42 | 0.45 |

**表 6** 交差検証誤差が最小のパラメータセットを用いた DBN, MLP, SVM の予測精度（182 次元の特徴ベクトルを用いた場合）

| | m300 | a300 | a600 | a1200 | a2400 | p300 | p600 |
|---|---|---|---|---|---|---|---|
| MLP 1 | 0.95 | 0.95 | 0.95 | 0.93 | 0.91 | **0.96** | 0.94 |
| MLP 2 | **0.96** | 0.95 | 0.94 | 0.93 | 0.92 | **0.96** | 0.95 |
| SVM (Linear) | 0.95 | 0.93 | 0.94 | 0.93 | 0.92 | 0.93 | 0.93 |
| SVM (RBF) | 0.92 | 0.95 | 0.91 | 0.95 | 0.92 | 0.86 | 0.89 |
| **DBN** | 0.95 | **0.96** | **0.97** | **0.97** | **0.95** | **0.96** | **0.98** |

| | p1200 | p2400 | a300p300 | a600p600 | a1200p1200 | a2400p2400 |
|---|---|---|---|---|---|---|
| MLP 1 | 0.94 | 0.93 | **0.96** | 0.95 | 0.96 | **0.97** |
| MLP 2 | **0.95** | 0.95 | 0.93 | 0.96 | 0.96 | 0.96 |
| SVM (Linear) | 0.93 | 0.93 | 0.92 | 0.94 | 0.94 | 0.95 |
| SVM (RBF) | 0.87 | 0.55 | 0.94 | 0.92 | **0.97** | 0.96 |
| **DBN** | 0.91 | **0.96** | **0.96** | **0.97** | 0.96 | **0.97** |

**表 7** 交差検証誤差を昇順に並べたときの上位 5 個のパラメータセットを用いた DBN, MLP, SVM の平均予測精度（182 次元の特徴ベクトルを用いた場合）

| | m300 | a300 | a600 | a1200 | a2400 | p300 | p600 |
|---|---|---|---|---|---|---|---|
| MLP 1 | 0.944 | 0.940 | 0.942 | 0.928 | 0.922 | 0.938 | 0.946 |
| LP 2 | 0.954 | 0.948 | 0.946 | 0.934 | 0.924 | **0.958** | 0.948 |
| SVM (Linear) | 0.950 | 0.930 | 0.942 | 0.928 | 0.920 | 0.930 | 0.930 |
| SVM (RBF) | 0.902 | 0.946 | 0.922 | 0.932 | 0.924 | 0.854 | 0.888 |
| **DBN** | **0.958** | **0.962** | **0.964** | **0.966** | **0.946** | 0.956 | **0.974** |

| | p1200 | p2400 | a300p300 | a600p600 | a1200p1200 | a2400p2400 |
|---|---|---|---|---|---|---|
| MLP 1 | 0.944 | 0.942 | 0.950 | 0.952 | 0.958 | 0.956 |
| MLP 2 | **0.954** | 0.948 | 0.932 | 0.960 | 0.958 | 0.960 |
| SVM (Linear) | 0.930 | 0.930 | 0.920 | 0.940 | 0.940 | 0.950 |
| SVM (RBF) | 0.834 | 0.686 | 0.944 | 0.920 | 0.964 | 0.956 |
| **DBN** | 0.944 | **0.950** | **0.958** | **0.970** | **0.966** | **0.968** |

下げてしまった．次に，ほとんどの場合において，ベースラインの予測精度は機械学習のそれよりかなり低かった．また，ほとんどの場合において，DBN がすべての機械学習において最高の予測精度を出している（各学習セットにおいて各機械学習手法中の最高の精度は太字で表されている）．

#### 4.2.2 223 次元の特徴ベクトルを使用した場合

前節の実験結果はすでに提案手法の予測精度が従来の機械学習手法より高いことを示しただけでなく，学習データにおけるノイズに対する頑健性もある程度示せたと考える．しかし上記

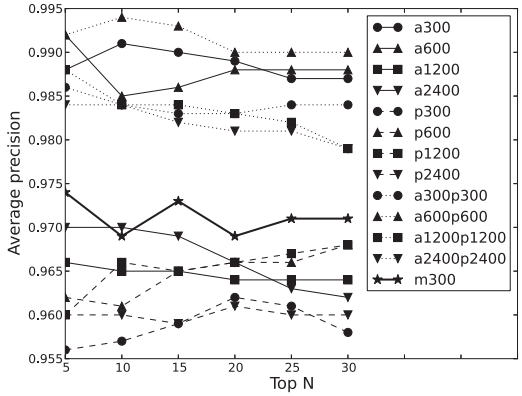
**表 8**　交差検証誤差を昇順に並べたときの上位 10 個のパラメータセットを用いた DBN, MLP, SVM の平均予測精度（182 次元の特徴ベクトルを用いた場合）

| | m300 | a300 | a600 | a1200 | a2400 | p300 | p600 |
|---|---|---|---|---|---|---|---|
| MLP 1 | 0.945 | 0.932 | 0.939 | 0.931 | 0.914 | 0.942 | 0.951 |
| MLP 2 | 0.951 | 0.944 | 0.943 | 0.933 | 0.924 | **0.954** | 0.953 |
| SVM (Linear) | 0.950 | 0.930 | 0.942 | 0.927 | 0.921 | 0.930 | 0.930 |
| SVM (RBF) | 0.960 | 0.941 | 0.914 | 0.936 | 0.924 | 0.842 | 0.872 |
| **DBN** | **0.961** | **0.962** | **0.965** | **0.968** | **0.948** | 0.948 | **0.964** |

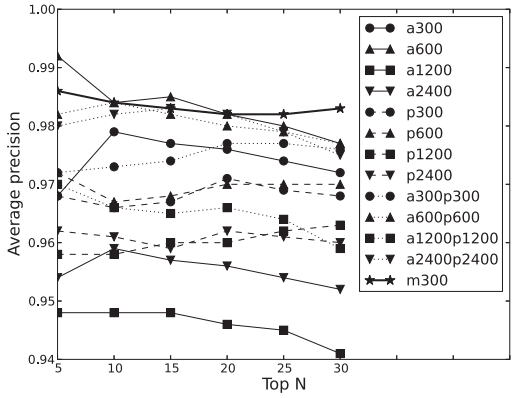
| | p1200 | p2400 | a300p300 | a600p600 | a1200p1200 | a2400p2400 |
|---|---|---|---|---|---|---|
| MLP 1 | 0.944 | 0.942 | 0.945 | 0.952 | 0.957 | 0.956 |
| MLP 2 | 0.952 | 0.949 | 0.941 | 0.955 | 0.958 | 0.961 |
| SVM (Linear) | 0.930 | 0.930 | 0.926 | 0.938 | 0.940 | 0.950 |
| SVM (RBF) | 0.822 | 0.757 | 0.936 | 0.926 | 0.952 | 0.951 |
| **DBN** | **0.954** | **0.950** | **0.953** | **0.961** | **0.963** | **0.968** |

実験では，手動学習データのラベル間の重複単語を除外していたため，疑似データの作成時はそれらをノイズとして加えることができず，提案手法のノイズへの頑健性に疑問が残る．本節の実験は，2 ラベル間の重複単語を残しているため，前節の実験よりも，より適切にノイズの頑健性を確認できると考える．

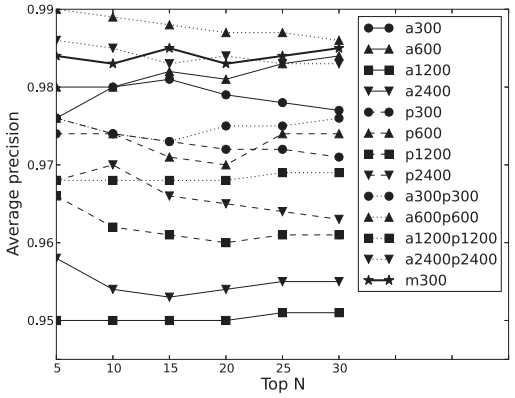
図 8 は図 6 と同様，各機械学習において，異なる学習データセットを用いた場合の評価データへの予測精度を示す．DBN のグラフにおいて，すべての点線と 2 本の実線が星マークの太線（つまり手動データ）の上にあること，また，SVM (RBF) においてすべての点線と実線が星マークの太線の上にあることから，前の実験結果と同様，DBN と SVM (RBF) については疑似データを含めたノイズのある学習データの利用が有効であることが確認できる．一方，MLP と SVM (Linear) については，手動データの星マークの太線がほとんど一番上に位置していることから，疑似データを含めたノイズのある学習データの有効性がほとんど見られない．すなわち，MLP と SVM (Linear) のノイズに対する頑健性については，前節の実験結果よりも悪い結果となった（逆に DBN の優位性がより顕著になったとも言える）．なお，182 次元の特徴ベクトルを用いた実験結果では a2400p2400 を用いた場合（逆三角形マークの点線），すなわち手動収集データに自動収集データと疑似データの両方を最も多く加えた場合，DBN が最高の精度を出しているのに対し，本実験結果では DBN は a600p600 を用いた場合（正三角形マークの点線）に最高の精度を出している．これは，精度の高いデータに対し，加えてよいノイズのあるデータについては適正の数があるはずで，次元数が増えると個々の特徴ベクトルの本来のノイズの度合いが増強したため，ノイズデータの適正数が減少したと考えることができ，両者の結果は矛盾しないと思われる．



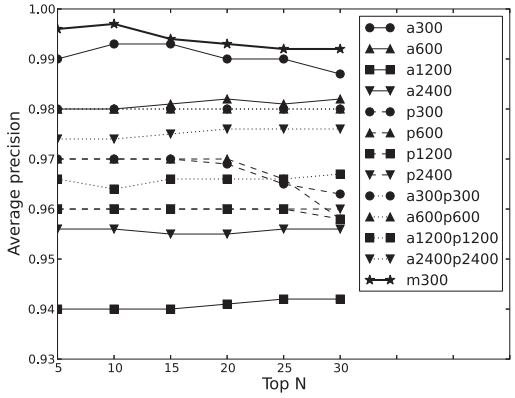

DBN



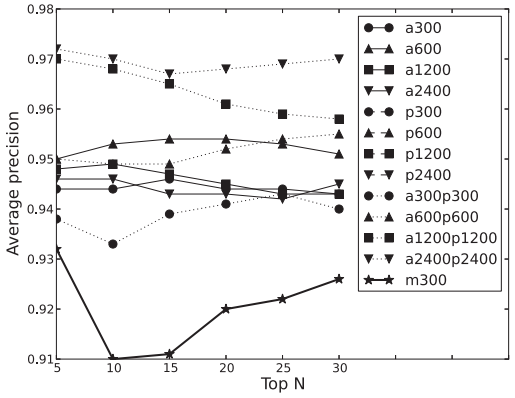

MLP 1




MLP 2




SVM (Linear)




SVM (RBF)

**図 8**　交差検証誤差の昇順で上位 N(5〜30) セットのパラメータを用いた場合の各機械学習の平均精度についての学習データセット間の比較（223 次元の特徴ベクトルを用いた場合）

### 4.2.3 有意差検定

交差検証誤差が最小のパラメータセットを用いた場合と，交差検証誤差を昇順に並べたときの上位 10 個のパラメータセットを用いた場合について，DBN と他の手法との性能の有意差検定を行った．交差検証誤差が最小のパラメータセットを用いた場合，各学習データセットについて単独で検定を行うとデータ数が少なすぎるため，各学習データセットの結果を 1 つにまとめて符号検定と t 検定を行った．一方，交差検証誤差上位 10 個のパラメータセットを用いた場合は各学習データセットについて単独で t 検定を行った．検定結果を表 9，10 に示す．これらの結

**表 9** 交差検証誤差が最小のパラメータセットを用いた場合の DBN と他の手法との性能比較に関する両側符号検定と t 検定の結果

| | 182 次元 | | 223 次元 | |
|---|---|---|---|---|
| | 符号検定 | t 検定 | 符号検定 | t 検定 |
| MLP 1 | 0.012 ** | 0.0441 ** | 0.037 ** | 0.0327 ** |
| MLP 2 | 0.082 * | 0.074 * | 0.123 | 0.1149 |
| SVM (Linear) | 0.000 *** | 0.0002 *** | 0.018 ** | 0.0763 * |
| SVM (RBF) | 0.000 *** | 0.0473 ** | 0.000 *** | 0.0075 *** |

数値は p 値であり，有意水準 10%で有意に差があるものには*，有意水準 5%で有意に差があるものには**，有意水準 1%で有意に差があるものには***を付けている．

**表 10** 各学習データセットについて，それらにおける交差検証誤差を昇順に並べたときの上位 10 個のパラメータセットを用いた場合の平均予測精度についての DBN と他の手法との性能比較に関する両側 t 検定の結果

| | 182 次元 | | | | 223 次元 | | | |
|---|---|---|---|---|---|---|---|---|
| | MLP 1 | MLP 2 | SVM (Linear) | SVM (RBF) | MLP 1 | MLP 2 | SVM (Linear) | SVM (RBF) |
| m300 | 0.005 *** | 0.237 | 0.057 * | 0.000 *** | 0.015 ** | 0.127 | 0.000 *** | 0.000 *** |
| a300 | 0.002 *** | 0.008 *** | 0.000 *** | 0.000 *** | 0.044 ** | 0.057 * | 0.555 | 0.000 *** |
| a600 | 0.000 *** | 0.000 *** | 0.000 *** | 0.000 *** | 0.798 | 0.244 | 0.138 | 0.000 *** |
| a1200 | 0.000 *** | 0.000 *** | 0.000 *** | 0.000 *** | 0.031 ** | 0.034 ** | 0.000 *** | 0.045 ** |
| a2400 | 0.001 *** | 0.000 *** | 0.000 *** | 0.000 *** | 0.032 ** | 0.000 *** | 0.000 *** | 0.000 *** |
| p300 | 0.329 | 0.297 | 0.007 *** | 0.000 *** | 0.225 | 0.028 ** | 0.057 * | 0.001 *** |
| p600 | 0.122 | 0.200 | 0.000 *** | 0.000 *** | 0.193 | 0.009 *** | 0.001 *** | 0.000 *** |
| p1200 | 0.221 | 0.808 | 0.008 *** | 0.000 *** | 0.235 | 0.653 | 0.313 | 0.000 *** |
| p2400 | 0.280 | 0.893 | 0.008 *** | 0.001 *** | 0.840 | 0.052 * | 1.000 | 0.000 *** |
| a300p300 | 0.269 | 0.154 | 0.003 *** | 0.035 ** | 0.017 ** | 0.032 ** | 0.104 | 0.000 *** |
| a600p600 | 0.159 | 0.239 | 0.005 *** | 0.000 *** | 0.008 *** | 0.096 * | 0.000 *** | 0.000 *** |
| a1200p1200 | 0.217 | 0.177 | 0.000 *** | 0.048 ** | 0.000 *** | 0.003 *** | 0.000 *** | 0.000 *** |
| a2400p2400 | 0.005 *** | 0.001 *** | 0.000 *** | 0.001 *** | 0.168 | 0.591 | 0.004 *** | 0.010 *** |

数値は p 値であり，有意水準 10%で有意に差があるものには*，有意水準 5%で有意に差があるものには**，有意水準 1%で有意に差があるものには***を付けている．

**表 11** 各学習データセットに対して交差検証誤差が最小のパラメータセットを用いた場合のラベルごとの全学習データセットにおける平均予測精度

| | 182 次元 | | | | | 223 次元 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MLP 1 | MLP 2 | SVM (Linear) | SVM (RBF) | DBN | MLP 1 | MLP 2 | SVM (Linear) | SVM (RBF) | DBN |
| CPU | 1.0 | 0.992 | 1.0 | 0.915 | 1.0 | 0.985 | 0.962 | 0.985 | 0.862 | 0.946 |
| グラフィックボード | 1.0 | 0.985 | 0.992 | 0.954 | 1.0 | 1.0 | 1.0 | 1.0 | 0.938 | 0.992 |
| ハードディスク | 0.938 | 0.946 | 0.915 | 0.908 | 0.923 | 0.962 | 0.977 | 0.977 | 0.938 | 0.969 |
| メインメモリ | 0.962 | 0.969 | 0.877 | 0.831 | 0.985 | 0.969 | 0.954 | 0.954 | 0.862 | 0.969 |
| マザーボード | 0.900 | 0.900 | 0.923 | 0.823 | 0.977 | 0.931 | 0.908 | 0.938 | 0.900 | 0.962 |
| OS | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.992 | 1.0 | 1.0 |
| 光学ドライブ | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.954 | 0.992 |
| PC ケース | 0.662 | 0.692 | 0.631 | 0.638 | 0.708 | 0.877 | 0.915 | 0.877 | 0.769 | 0.977 |
| 電源ユニット | 1.0 | 1.0 | 1.0 | 0.885 | 1.0 | 0.954 | 0.977 | 0.985 | 0.846 | 1.0 |
| SSD | 1.0 | 0.992 | 1.0 | 0.977 | 1.0 | 1.0 | 1.0 | 1.0 | 0.938 | 0.985 |

果から，182 次元と 223 次元の特徴ベクトルのいずれを用いても，多数の場合において DBN が他の手法より有意に優れていることが確認できる．また，詳細をみると，たとえば a2400p2400 の学習データセットについては 182 次元の特徴ベクトルを，a600p600/a1200p1200 の学習データセットについては 223 次元の特徴ベクトルを用いたほうが有意差が顕著であることがわかり，特徴ベクトルの構成方法について，DBN と他の手法との性能差の観点からどれが一番よいかは一概に断言することができない[15]．

最後に，参考として，各手法のラベル（検索語）ごとの予測精度（表 11）と，交差検証誤差が最小のパラメータセット（表 12）を示しておく．表 11 から，182 次元の「PC ケース」を除き各ラベルへの予測精度にばらつきが小さいことがわかる．また，全般的に DBN のほうがほかの手法より各ラベルに対する予測精度がよいことがわかる．さらに，たとえば DBN の予測精度は 182 次元の場合のほうが 10 個中の 6 個のラベルについて 223 次元の場合に勝っており，182 次元と 223 次元のどちらのほうがよいかが一概に言えないことがわかる．表 12 には，隠れ層のユニット数が 182 次元で 273，223 次元で 335 が多く出現しており，隠れ層のユニット数は多いほうがよいという Bengio の提言と合致している．

## 5 本課題の意義について

本研究では特定の分野の関連語・周辺語または説明文書を入力としたときの検索用語の予

[15] この結果については 4.2.2 節でも述べたように，ノイズデータについては適正の数があるはずであることと，次元数が増えると個々のベクトルの本来のノイズの度合いが増強する（よってノイズデータの適正数が減る）ことを合わせて考えれば両者の結果は矛盾しないと思われる．

**表 12**　DBN の各学習データセットに対して交差検証誤差が最小のパラメータセット

| | 182 次元 | | | | | 223 次元 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Pre-training | | Fine-training | | 隠れ層 | Pre-training | | Fine-training | | 隠れ層 |
| | 学習率 | 学習回数 | 学習率 | 学習回数 | | 学習率 | 学習回数 | 学習率 | 学習回数 | |
| m300 | 0.1 | 3,000 | 0.1 | 500 | 91 | 0.1 | 3,000 | 0.1 | 3,000 | 167-112 |
| a300 | 0.1 | 3,000 | 0.01 | 2,000 | 273 | 0.1 | 3,000 | 0.01 | 3,000 | 335 |
| a600 | 0.1 | 1,000 | 0.1 | 1,000 | 273 | 0.1 | 2,000 | 0.01 | 3,000 | 335 |
| a1200 | 0.1 | 1,000 | 0.1 | 1,000 | 273 | 0.1 | 2,000 | 0.1 | 1,000 | 335 |
| a2400 | 0.1 | 3,000 | 0.1 | 2,000 | 273 | 0.1 | 3,000 | 0.1 | 2,000 | 335 |
| p300 | 0.01 | 3,000 | 0.1 | 3,000 | 91 | 0.1 | 2,000 | 0.1 | 3,000 | 186-149-112 |
| p600 | 0.1 | 1,000 | 0.1 | 2,000 | 273 | 0.01 | 1,000 | 0.1 | 3,000 | 335 |
| p1200 | 0.001 | 1,000 | 0.1 | 3,000 | 152-121-91 | 0.01 | 2,000 | 0.1 | 3,000 | 335 |
| p2400 | 0.1 | 3,000 | 0.1 | 3,000 | 273 | 0.1 | 3,000 | 0.1 | 3,000 | 335 |
| a300p300 | 0.1 | 500 | 0.1 | 3,000 | 91 | 0.1 | 500 | 0.01 | 3,000 | 112 |
| a600p600 | 0.1 | 1,000 | 0.1 | 2,000 | 273 | 0.1 | 2,000 | 0.1 | 3,000 | 335 |
| a1200p1200 | 0.1 | 1,000 | 0.1 | 3,000 | 273 | 0.1 | 3,000 | 0.1 | 2,000 | 335 |
| a2400p2400 | 0.1 | 2,000 | 0.1 | 3,000 | 273 | 0.1 | 2,000 | 0.1 | 3,000 | 335 |

測・提示を行う検索支援を想定している．まず，説明文書による支援の意義は，たとえば The 5th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access のようなワークショップ型共同研究 (Ma, Nakao, and Murata 2005) における長い文書を検索課題[16]としたタスクからも類推できる．つまり，たとえばユーザが関連語・周辺語もはっきりわからないときはその支援要求を文書の形で伝える（入力する）ニーズはあると考える．また一方，当然のことではあるが，本研究では，少数キーワード（関連語・周辺語）による検索用語の予測も期待している．実際，表 14 は，DBN について，各学習データセットを用いた場合の，表 13 に示す 3 関連語・周辺語（+1 ノイズ語）[17]による全検索用語の平均予測精度を示している[18]．実験はまだ小規模ではあるが，この結果は提案手法が少数キーワードによる支援も可能であることを示唆していると思われる．

## 6　結び

本稿では深層学習の代表的な手法である Deep Belief Network (DBN) を用いて関連語・周辺

[16] 検索課題例：AOL とタイムワーナー合併の影響に関する記事を探したい．AOL・タイムワーナー合併がインターネットとエンターテインメントというメディア産業に与える影響に関する意見を適合とする．AOL・タイムワーナー合併の展開についての記述は部分的に適合とする．総額と所有権転換の仕組みに関する情報は不適合とする．

[17] これらのキーワードは予備実験も含め一切精査せずに著者らの知識に頼って手動収集のデータから関連語・周辺語・ノイズ語としてふさわしいものを主観で選んでいる．しかし当然なことではあるが，これらのキーワードはすべて誰にも知られている用語である保証はない（また，本実験の目的からしてそう保証する必要もない）．

[18] 当然のことではあるが，予測精度は用いるキーワードに大きく依存する．試しに 10 検索用語のうち 6 検索用語の関連語・周辺語を意識的に関連性の弱いものを選んで実験すると平均精度が 8 割程度までに下がった．

**表 13** 予測に用いるキーワード

| | 関連語・周辺語 | ノイズ語 |
|---|---|---|
| CPU | 頭脳，計算，コア | 管理 |
| グラフィックボード | 映像，ディスプレイ，描画 | デザイン |
| ハードディスク | 磁気ヘッド，円盤，プラッタ | 読み込み |
| メインメモリ | 作業，処理速度，アクセス | メディア |
| マザーボード | 基盤，ソケット，チップセット | 頭脳 |
| OS | 管理，Windows, 基本ソフトウェア | 計算 |
| 光学ドライブ | メディア，再生，レーザー | 円盤 |
| PC ケース | 箱，デザイン，収納 | コンセント |
| 電源ユニット | 供給，電圧，変換 | 箱 |
| SSD | 衝撃，フラッシュメモリ，振動 | プラッタ |

**表 14** DBN の少数キーワードによる予測精度（223 次元の特徴ベクトルを用いた場合）

| | ノイズ語なし | ノイズ語あり |
|---|---|---|
| m300 | 1.0 | 0.9 |
| a300 | 1.0 | 0.9 |
| a600 | 0.9 | 0.8 |
| a1200 | 1.0 | 0.8 |
| a2400 | 0.9 | 0.7 |
| p300 | 1.0 | 1.0 |
| p600 | 1.0 | 1.0 |
| p1200 | 1.0 | 1.0 |
| p2400 | 1.0 | 1.0 |
| a300p300 | 1.0 | 1.0 |
| a600p600 | 1.0 | 1.0 |
| a1200p1200 | 1.0 | 1.0 |
| a2400p2400 | 1.0 | 1.0 |
| 平均 | 0.985 | 0.931 |

語またはそれらの語から構成される説明文書から適切な検索用語を予測する手法を提案した．DBN の有効性を確認するために，用例に基づくベースライン手法，多層パーセプトロン (MLP)，およびサポートベクトルマシン (SVM) との比較を行った．学習と評価に用いるデータは手動と自動の 2 通りの方法でインターネットから収集した．加えて，自動生成した疑似データも用いた．各種機械学習の最適なパラメータはグリッドサーチと交差検証を行うことにより決めた．実験の結果，DBN の予測精度はベースライン手法よりはるかに高く MLP と SVM のいずれよりも高かった．また，手動収集データに自動収集のデータと疑似データを加えて学習することにより予測精度は向上した．さらに，よりノイズの多い学習データを加えても DBN の予測精度はさらに向上した．しかしながらこの場合 MLP の精度向上は見られなかった．このことか

ら，DBN のほうが MLP よりもノイズの多い学習データを有効利用できることが分かった．なお，まだ少数の実験例しかなかったが，提案手法が少数キーワードによる支援も可能であることを示唆した実験結果も得られた．

今後はより大規模な評価実験を通じ，提案手法の有効性の確認を行うとともに，様々な分野における実用的な検索用語の予測システムを構築していく予定である．

## 謝　辞

本稿に対して丁寧かつ有益なご意見ご指摘をいただきました査読者の方に感謝いたします．本稿の内容の一部は，The 28th Pacific Asia Conference on Language, Information and Computing (Paclic 28) で発表したものです (Ma, Tanigawa, and Murata 2014)．また，本研究は JSPS 科研費 25330368 の助成を受けています．記して謝意を表します．

## 参考文献

粟飯原俊介，長尾真，田中久美子 (2013). 意味的逆引き辞書『真言』. 言語処理学会第 19 回年次大会発表論文集, pp. 406–409.

Auli, M., Galley, M., Quirk, C., and Zweig, G. (2013). “Joint Language and Translation Modeling with Recurrent Neural Networks.” In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1044–1054.

Bengio, Y. (2009). “Learning Deep Architectures for AI.” *Foundations and Trends in Machine Learning*, **2** (1), pp. 1–127.

Bengio, Y. (2012). “Practical Recommendations for Gradient-Based Training of Deep Architectures.” *eprint arXiv:1206.5533*, pp. 1–33.

Bengio, Y., Courville, A., and Vincent, P. (2013). “Representation Learning: A Review and New Perspectives.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35** (8), pp. 1798–1828.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). “Greedy Layer-wise Training of Deep Networks.” In *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, pp. 153–160.

Billingsley, R. and Curran, J. (2012). “Improvements to Training an RNN Parser.” In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pp. 279–294.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). “Nat-

ural Language Processing (Almost) from Scratch." *Journal of Machine Learning Research*, **12**, pp. 2493–2537.

Glorot, X. and Bengio, Y. (2010). "Understanding the Difficulty of Training Deep Feedforward Neural Networks." In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, pp. 249–256.

Glorot, X., Bordes, A., and Bengio, Y. (2011). "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach." In *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, pp. 513–520.

Hashimoto, K., Miwa, M., Tsuruoka, Y., and Chikayama, T. (2013). "Simple Customization of Recursive Neural Networks for Semantic Relation Classification." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1372–1376.

Hermann, K. M. and Blunsom, P. (2013). "The Role of Syntax in Vector Space Models of Compositional Semantics." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 894–904.

Hiton, G. E., Osindero, S., and Teh, Y. (2006). "A Fast Learning Algorithm for Deep Belief Nets." *Neural Computation*, **18**, pp. 1527–1554.

Kalchbrenner, N. and Blunsom, P. (2013). "Recurrent Continuous Translation Models." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1700–1709.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pp. 1097–1105.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations." In *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*, pp. 609–616.

Li, L., Zhao, Y., Jiang, D., Zhang, Y., Wang, F., Gonzalez, I., Valentin, E., and Sahli, H. (2013). "Hybrid Deep Neural Network - Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition." In *Proceedings of Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII 2013)*, pp. 312–317.

Liu, L., Watanabe, T., Sumita, E., and Zhao, T. (2013). "Additive Neural Networks for Statistical Machine Translation." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 791–801.

Luong, T., Socher, R., and Manning, C. (2013). "Better Word Representations with Recur-

sive Neural Networks for Morphology." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 104–113.

Ma, Q., Nakao, K., and Murata, M. (2005). "Single Language Information Retrieval at NTCIR-5." In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, pp. 39–43.

Ma, Q., Tanigawa, I., and Murata, M. (2014). "Retrieval Term Prediction Using Deep Belief Networks." In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing (Paclic 28)*, pp. 338–347.

内木賢吾，佐藤理史，駒谷和範 (2013). 日本語クロスワードを解く：性能向上の検討. 2013 年度人工知能学会全国大会.

Salakhutdinov, R. and Hinton, G. E. (2009). "Semantic Hashing." *International Journal of Approximate Reasoning*, **50** (7), pp. 969–978.

Seide, F., Li, G., and Yu, D. (2011). "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." In *Proceedings of 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011)*, pp. 437–440.

Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y. (2013). "Parsing with Computational Vector Grammars." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 455–465.

Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., and Manning, C. D. (2011). "Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection." In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, pp. 801–809.

Socher, R., Perelygin, A., Wu, J. Y., and Chuang, J. (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1631–1642.

Srivastava, S., Hovy, D., and Hovy, E. H. (2013). "A Walk-Based Semantically Enriched Tree Kernel Over Distributed Word Representations." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1411–1416.

Tsubaki, M., Duh, K., Shimbo, M., and Matsumoto, Y. (2013). "Modeling and Learning Semantic Co-Compositionality through Prototype Projections and Neural Networks." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 130–140.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. A. (2008). "Extracting and Composing Robust Features with Denoising Autoencoders." In *Proceedings of the 25th International*

*Conference on Machine Learning (ICML 2008)*, pp. 1096–1103.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. A. (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion." *Journal of Machine Learning Research*, **11**, pp. 3371–3408.

Zou, W. Y., Socher, R., Cer, D. M., and Manning, C. D. (2013). "Bilingual Word Embeddings for Phrase-Based Machine Translation." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1393–1398.

## 略歴

**馬　　青**：1983 年北京航空航天大学自動制御学科卒業．1987 年筑波大学大学院修士課程理工学研究科理工学専攻修了．1990 年同大学院博士課程工学研究科電子・情報工学専攻修了．工学博士．郵政省通信総合研究所・独立行政法人通信総合研究所（現国立研究開発法人情報通信研究機構）主任研究官・主任研究員を経て，2003 年 4 月より，龍谷大学理工学部教授．自然言語処理，機械学習の研究に従事．言語処理学会，情報処理学会，電子情報通信学会，日本神経回路学会，各会員．

**谷河　息吹**：2013 年龍谷大学数理情報学科卒業．2015 年龍谷大学大学院理工学研究科数理情報学専攻修士課程修了．人工知能学会員．

**村田　真樹**：1993 年京都大学工学部電気工学第二学科卒業．1997 年同大学院工学研究科電子通信工学専攻博士課程修了．博士（工学）．同年，京都大学にて日本学術振興会リサーチ・アソシエイト．1998 年郵政省通信総合研究所入所．独立行政法人情報通信研究機構主任研究員を経て，2010 年 4 月より，鳥取大学大学院工学研究科情報エレクトロニクス専攻教授．自然言語処理，情報抽出の研究に従事．言語処理学会，情報処理学会，電子情報通信学会，人工知能学会，計量国語学会，各会員．

（2015 年 2 月 27 日　受付）
（2015 年 6 月 29 日　再受付）
（2015 年 8 月 12 日　採録）